

FMS: the Flexible Modeling System

SSAM-2012
GFDL, Princeton NJ

V. Balaji

`balaji@princeton.edu`

NOAA/GFDL and Princeton University

18 July 2012

Outline

1 FMS and FRE

- FMS: the Flexible Modeling System
- FRE: the FMS Runtime Environment

2 The FMS Coupler

- FMS Coupled Architecture
- Serial and concurrent coupling
- The Exchange Grid
- Mosaics

3 FMS/AM3 Code Architecture

4 FMS: Lessons from long use

- Strengths and weaknesses

Outline

- 1 FMS and FRE
 - FMS: the Flexible Modeling System
 - FRE: the FMS Runtime Environment

- 2 The FMS Coupler
 - FMS Coupled Architecture
 - Serial and concurrent coupling
 - The Exchange Grid
 - Mosaics

- 3 FMS/AM3 Code Architecture

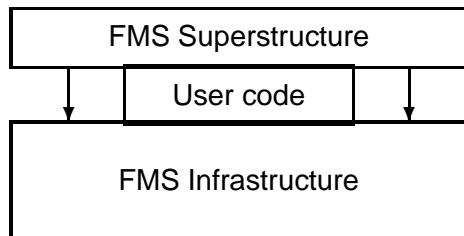
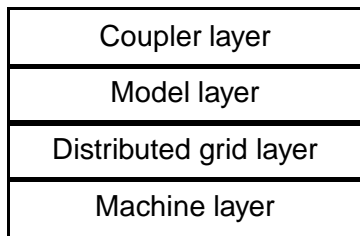
- 4 FMS: Lessons from long use
 - Strengths and weaknesses

"Milestones in Scientific Computing", from Nature (23 March 2006)



- 1946 "ENIAC, ... the first electronic digital computer"
- 1969 "results from first coupled ocean-atmosphere general circulation model are published by Syukuro Manabe and Kirk Bryan, paving the way for later climate simulations that become a powerful tool in research on global warming...."
- 1972 "... the first hand-held scientific calculator"
- 1989 "Tim Berners-Lee ... develops the World Wide Web"

FMS is in its second decade of active use



- Flexible Modeling System effort began in 1998, when GFDL first moved on to distributed memory machines
- Provided simplified interface to parallelism and I/O: `mp_p`. Abstract types for “distributed grid” and “fields”.
- Diagnostic output, data override, time manager.
- Component-based design, abstract types for component state vectors, exchange grid.
- “Sandwich” model influential in community.

FRE: the FMS Runtime Environment

- Operational since 2003, designed to provide an environment for integrated testing and production.
- Rigorous standardized test procedure for evaluating new code and new model assemblies and configurations.
- Integrated existing post-processing structure.
- Captures complete configuration from source assembly to compilation to running, post-processing and analysis.
- Simulation database provides retrieval of model output, model analysis, and now model state and configuration information.
- Again influential in community, with “curators” being prototyped at various sites.

Outline

- 1 FMS and FRE
 - FMS: the Flexible Modeling System
 - FRE: the FMS Runtime Environment

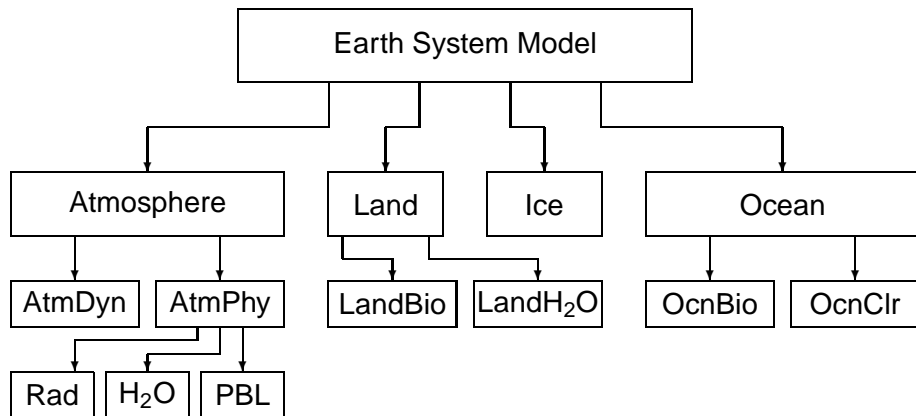
- 2 The FMS Coupler
 - FMS Coupled Architecture
 - Serial and concurrent coupling
 - The Exchange Grid
 - Mosaics

- 3 FMS/AM3 Code Architecture

- 4 FMS: Lessons from long use
 - Strengths and weaknesses

FMS Physical Architecture

Beyond a common modeling architecture, FMS offers a common physical architecture.



The FMS coupler

Used for data exchange between models. Key features include:

Conservation: required for long runs.

Resolution: no constraints on component model timesteps and spatial grid. Supports both explicit and implicit timestepping.

Exchange grid: union of component model grids, where detailed flux computations are performed (Monin-Obukhov, tridiagonal solver for implicit diffusion, ...)

Fully parallel: Calls are entirely processor-local: exchange software will perform all inter-processor communication.

Single executable: serial and concurrent execution in a single executable.

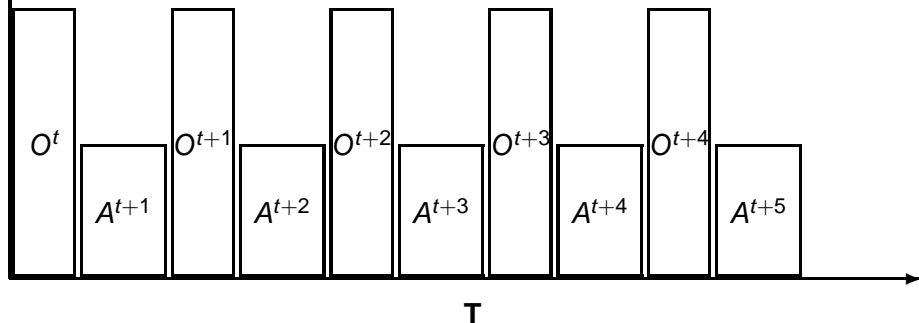
Highly efficient: currently able to couple atmos/ocean explicitly at each ocean timestep, atmos/land/ice implicitly at each atmos timestep for current dec/cen models.

Serial coupling

Uses a forward-backward timestep for coupling.

$$A^{t+1} = A^t + f(O^t) \quad (1)$$

$$O^{t+1} = O^t + f(A^{t+1}) \quad (2)$$

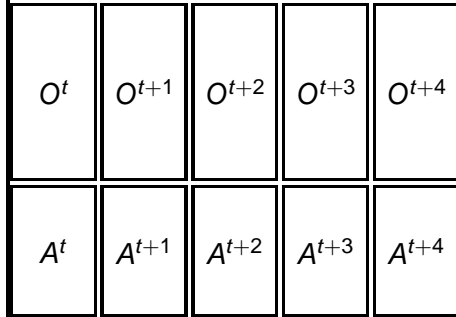


Concurrent coupling

This uses a forward-only timestep for coupling. While formally this is unconditionally unstable, the system is strongly damped. Answers vary with respect to serial coupling, as the ocean is now forced by atmospheric state from Δt ago.

$$A^{t+1} = A^t + f(O^t) \quad (3)$$

$$O^{t+1} = O^t + f(A^t) \quad (4)$$



Implicit coupling and the exchange grid

Fluxes at the surface often need to be treated using an implicit timestep. (e.g temperature flux in near-surface layers that can have vanishingly small heat capacity.)

$$\frac{\partial T}{\partial t} = K \frac{\partial^2 T}{\partial z^2} \quad (5)$$

$$\frac{T_k^{n+1} - T_k^n}{\Delta t} = K \frac{T_{k+1}^{n+1} + T_{k-1}^{n+1} - 2T_k^{n+1}}{\Delta z^2} \quad (6)$$

$$\mathbf{A} \mathbf{T}^{n+1} = \mathbf{T}^n \quad (7)$$

Implicit coupling and the exchange grid

Tridiagonal solver in Eq. 7 across multiple components and grids.

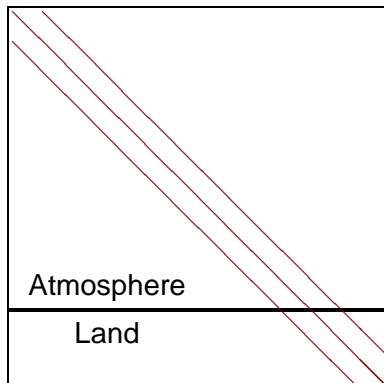
Atmosphere



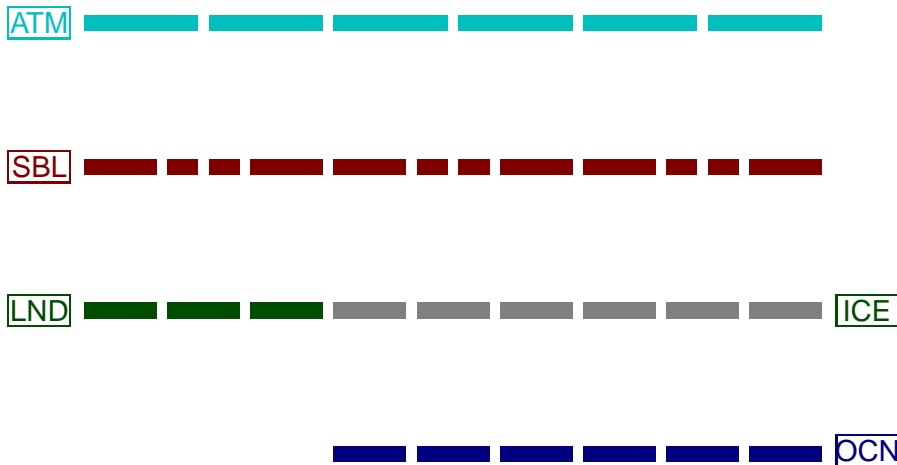
Exchange



Land



Coupled architecture with SBL on exchange grid



Flux exchange

Three types of flux exchange are permitted: REGRID, REDIST and DIRECT.

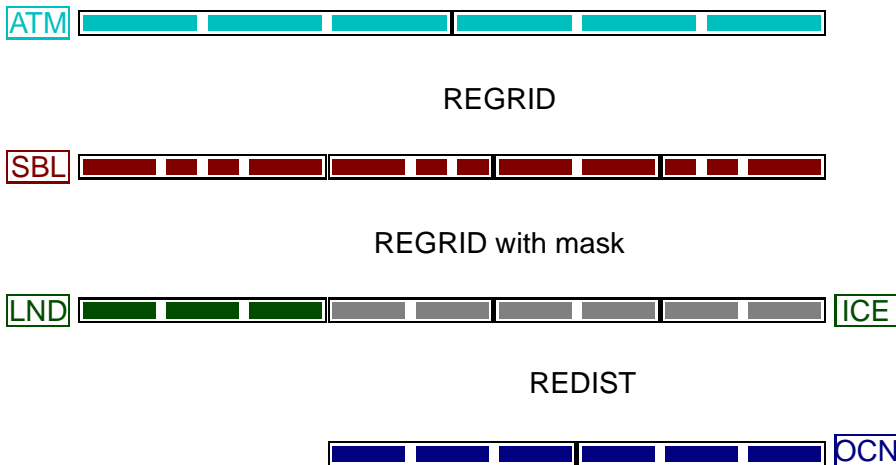
REGRID physically distinct grids, requires exchange grid.

REDIST identical global grid, different domain decomposition.

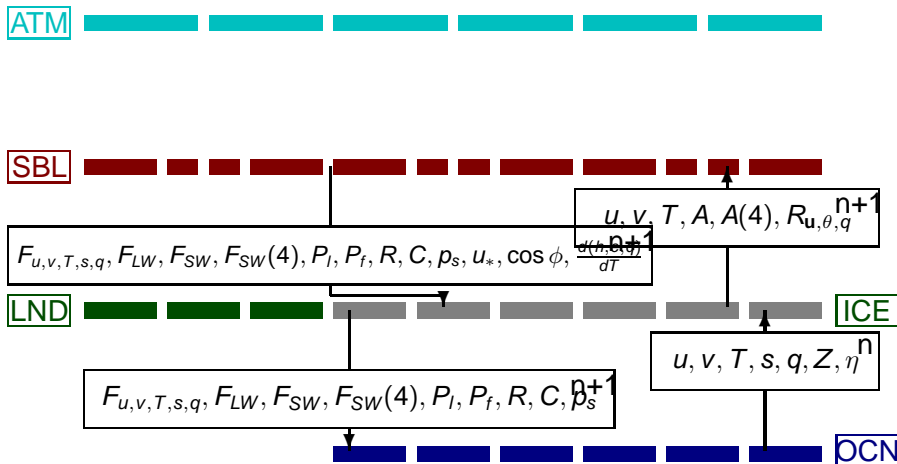
DIRECT identical grid and decomposition.

Current use: REGRID between $\text{atmos} \longleftrightarrow \text{ice}$, $\text{atmos} \longleftrightarrow \text{land}$, $\text{land} \longleftrightarrow \text{ice}$, REDIST between $\text{ocean} \longleftrightarrow \text{ice}$.

Parallelism in the FMS coupler



FMS coupled architecture: ice-ocean coupling



Exchange grid: features

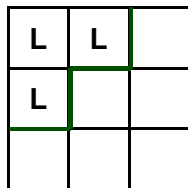
- Each cell on exchange grid “belongs” to one cell on each parent grid;
- Conservative interpolation up to second order;
- All calls exchange local data; data-sharing among processors is internal to the exchange software, and non-blocking.
- Physically identical grids (e.g ocean and sea ice) exchange data without interpolation.
- Exchange grid is computed and stored offline following a **gridspec** netCDF “standard”.

Exchange grid size

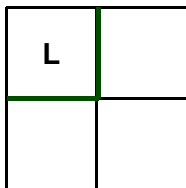
Atmosphere	Ocean	Xgrid	Density	Scalability
144×90	360×200	79644	8.5×10^{-5}	0.29
288×180	1080×840	895390	1.9×10^{-5}	0.56

Exchange grid sizes for typical climate model grids. The first column shows the horizontal discretization of an atmospheric model at “typical” climate resolutions of 2° and 1° respectively. The `ocean` column shows the same for an ocean model, at 1° and $\frac{1}{3}^\circ$. The `xgrid` column shows the number of points in the computed exchange grid, and the density relates that to the theoretical maximum number of exchange grid cells. The `scalability` column shows the load imbalance of the exchange grid relative to the overall model when it inherits its parallel decomposition from one of the parent grids.

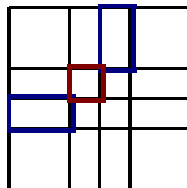
The mask problem



Land



Ocean



Exchange

An issue arises when grids of two independent components (e.g land and sea) share a boundary. The boundary is defined by a **mask** (e.g land-sea mask) but the mask is discretized independently on the two grids. However, exchange grid cells need to be uniquely assigned to a single component. This means that some cells get **clipped** on one or the other grid. In FMS, by convention, we choose to clip the land grid.

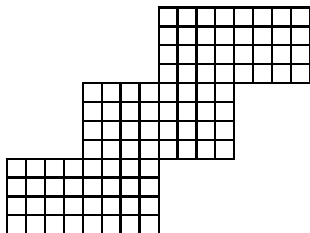
Exchange grid: literature

- Balaji *et al* 2006: “The FMS Exchange Grid: a mechanism for data exchange between Earth System components on independent grids”:

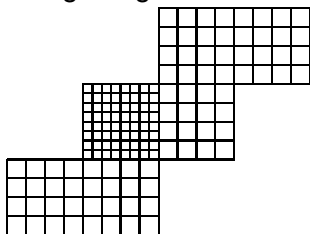
<http://www.gfdl.noaa.gov/~vb/pdf/xgridpaper.pdf>

- Isaac Held's notes: “Surface Fluxes, Implicit Time Stepping, and the Exchange Grid: The Structure of the Surface Exchange Module” <http://www.gfdl.noaa.gov/~vb/pdf/surf.pdf>

What is a grid mosaic?



Regular grid mosaic.



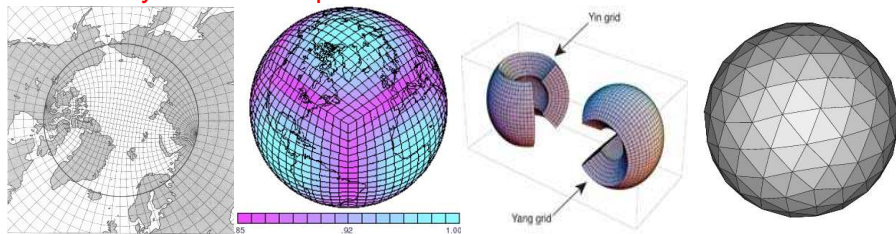
Refined grid mosaic.

Much current software only supports what we call *grid tiles* here. The `grid mosaic` extension will allow the development of more complex grids for models (e.g cubed-sphere). Further developments will include support for irregular tiling (e.g of the ocean surface following coastlines), and for refined, nested and adaptive grids.

Also, regular grids where an irregular decomposition is needed (e.g for a polar filter) can use mosaics to define different decompositions in different regions.

Mosaic grid specification

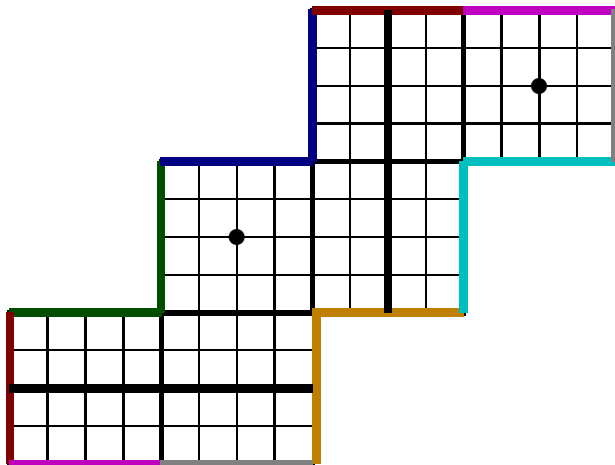
Grid mosaics can be composed of single tiles, or multiple tiles sharing a **boundary** or an **overlap** contact region



Boundary spec for a cubed sphere

Boundaries for LRG tiles are specified in terms of an **anchor point** and an **orientation**.

The orientation of the boundary specifies the index space direction of the running boundary on each grid tile: the point just to the “west” of $(5, 6)$ is in fact $(3, 4)$



Applications of grid mosaics

The grid mosaic is a powerful abstraction making possible an entire panoply of applications. These include:

- the use of overset grids such as the yin-yang grid;
- the representation of nested grids (e.g Kurihara et al 1990);
- the representation of reduced grids (e.g Rasch 1994). Currently these typically use full arrays and a specification of the “ragged edge”.
- An entire coupled model application or dataset can be constructed as a hierarchical mosaic.
- Finally, grid mosaics can be used to overcome performance bottlenecks associated with parallel I/O and very large files.

Outline

- 1 FMS and FRE
 - FMS: the Flexible Modeling System
 - FRE: the FMS Runtime Environment

- 2 The FMS Coupler
 - FMS Coupled Architecture
 - Serial and concurrent coupling
 - The Exchange Grid
 - Mosaics

- 3 FMS/AM3 Code Architecture

- 4 FMS: Lessons from long use
 - Strengths and weaknesses

coupler_main slow loop

From `src/coupler/coupler_main.F90`:

```
do nc = 1, num_cpld_calls
  call flux_ocean_to_ice( Ocean, Ice, Ocean_ice_flux )
  call update_ice_model_slow_up( Ocean_ice_flux, Ice )
!fast loop
  call update_land_model_slow(Land)
  call flux_land_to_ice( Land, Ice, Land_ice_flux )
  call update_ice_model_slow_dn( Atmos_ice_flux, Land_ice_flux, Ice )
  call flux_ice_to_ocean( Ice, Ice_ocean_flux )
  call update_ocean_model( Ice_ocean_flux, Ocean )
enddo
```

(8)

Types are defined in the parent models.

From `src/coupler/coupler_main.F90`:

```
do na = 1, num_atmos_calls
  Time = Time + Time_step_atmos
  call sfc_boundary_layer( Atm, Land, Ice, &
    Land_ice_atmos_flux )
  call update_atmos_model_down( Land_ice_atmos_flux, Atm )
  call flux_down_from_atmos( Time, Atm, Land, Ice, &
    Land_ice_atmos_flux, &
    Atmos_land_flux, Atmos_ice_flux )
  call update_land_model_fast( Atmos_land_flux, Land )
  call update_ice_model_fast( Atmos_ice_flux, Ice )
  call flux_up_to_atmos( Time, Land, Ice, Land_ice_atmos_flux )
  call update_atmos_model_up( Land_ice_atmos_flux, Atm )
enddo
```

(9)

AM3 code architecture

(Source directories `atmos_shared`, `atmos_coupled`, `atmos_fv_dynamics`, `atmos_param`)

- `atmos_model_init`: initialization
- `update_atmos_model_down`: update dynamical state (`fv_dynamics`), advect tracers, then compute radiative transfer
- `update_atmos_model_up`: after completion of surface fluxes, update atmospheric thermodynamics including adjustments from moist processes (PBL, clouds and convection).
- `atmos_model_end`: write output and exit.

Atmospheric physics is all column-organized (dependencies in `k` only). Columns are organized into `windows` which can run in parallel.

Outline

- 1 FMS and FRE
 - FMS: the Flexible Modeling System
 - FRE: the FMS Runtime Environment

- 2 The FMS Coupler
 - FMS Coupled Architecture
 - Serial and concurrent coupling
 - The Exchange Grid
 - Mosaics

- 3 FMS/AM3 Code Architecture

- 4 FMS: Lessons from long use
 - Strengths and weaknesses

Our strengths in retrospect

- Scalable high-performance framework on up to $\mathcal{O}(10^5)$ processors.
- Good, stable, dedicated team in Modeling Services.
- Broad acceptance and widespread contributions to a working system: many useful contributions from external users.
- Impressive list of features: mosaics, parallel ensemble capability, experiment database. Equally impressive list of components and options.
- Agile: CM2.0 \rightarrow CM2.1 in ~ 2 months. Framework too: see `mpp` changes for PSETs and XT.
- Never scored high on Top500 or in flop-counting exercises: system balance and real performance always comes first.
- “Unprincipled” use of software engineering principles.
- Good balance between openness and realism about new technologies.
- “Rough consensus and working code”. (IETF)

Our weaknesses

- Component list:
 - atmosphere dycore: FV-CS, FV-LL, BGRID, SPECTRAL, ZETAC, data, null.
 - atmospheric physics and chemistry: AM2, AM2.1 (HiRAM), AM2.1 (Lin), AM3, simple, dry.
 - ocean: GOLD, MOM4p1, MOM4p0, mixed-layer, data, null.
 - land: LAD/LM2, SHE/LM3v, LAD2/LM3.
 - ocean BGC: TOPAZ, COBALT, BLING.

Almost every combinatoric possibility is in active use.

- Use every single F90 feature in the manual.
- FRE fails at fuzzy boundaries between model code, input, and configuration.
- Unsupported legacy code.
- Lack of consensus in lab about outside users and outside code.

Thank you! Questions?

